



HOKKAIDO UNIVERSITY

| | |
|------------------|---|
| Title | FOTRANによる英文和訳 |
| Author(s) | 中村, 忠之; Nakamura, Tadayuki; 森, 伸夫 他 |
| Citation | 北海道大學工學部研究報告, 59, 65-73 |
| Issue Date | 1971-03-10 |
| Doc URL | https://hdl.handle.net/2115/41026 |
| Type | departmental bulletin paper |
| File Information | 59_65-74.pdf |



FÖRTRAN による英文和訳

中 村 忠 之* 森 伸 夫**
須 戸 満** 栃 内 香 次**
仲 丸 由 正**

(昭和45年9月30日受理)

A FORTRAN Program for English-Japanese Translation

Tadayuki NAKAMURA Nobuo MORI
Mitsuru SUDO Koji TOCHINAI
Yoshimasa NAKAMARU
(Received September 30, 1970)

Abstract

In this report we describe an experimental computer program written in FORTRAN language for the translation of English into Japanese.

The program of the "unpack" operation which divides the contents of one word into words, or the "pack" operation which gathers the contents of some words into one word, can not be written directly in FORTRAN. Accordingly, we developed programs executing such operations by the use of exponentiation and have completed the program with FORTRAN.

We also compared the execution time and the length of the program written in FORTRAN only, with those of the program using machine language for the unpack or pack operation, and we may conclude that the former is more practical.

The method of syntactic analysis used here, is based on the immediate constituent theory, and in this experimental program we used 160 English words, 190 Japanese equivalents, and about 120 gramatical patterns. There are 1090 computer words used in this whole program, and it seems to be but a small number considering the complexity of processing.

1. 序 言

一般に、言語翻訳のように文字データを電子計算機で処理する場合には、プログラムは機械語かアセンブラー語で書かれているのが普通であり、広く用いられていて他機種と互換性のある FÖRTRAN 言語は不向きである。これは FÖRTRAN には計算機内部の1語をビットやキャラクター単位で扱うステートメントがないこと、ある語を分割したり、逆に数語にわたるデータを一つの情報として扱うというステートメントも考えられていないためである。機械翻訳のプログラムにはどうしても上述の機能が必要であり、それらの部分だけ機械語を用いる方法も考えられるが、べき乗の計算を利用することにより、同じ機能を持たせることができるということがわかったので、すべて FÖRTRAN で英文和訳のプログラムを作成したのでここに報告する。

FÖRTRAN プログラミングの利点は、プログラミングが容易であることと、他機種とし互換性が十分保たれることである。したがってプログラミングに際しては、機械翻訳の機能を失わず

* 日本アイ・ビー・エム株式会社

** 電子工学科 電子機器工学講座

できるだけ簡潔なものとする、および他機種との互換性をできるだけ保つことに重点をおいた。また、翻訳の対象となる言語データの表現も人間にわかりやすいということとともに、FÖRTRAN で処理しやすいように考慮してある。

2. 機械翻訳の方法

図1に機械翻訳の概略の手順を示してある。まず原文を読み込み、次に単語の検索を行なう。ここで問題となるのは大量のデータをいかにしてコンパクトに記憶装置内に格納するか、また検索の時間をいかにして縮少するかである。英単語のように文字の長さの異なるものを同類のデータとして処理するには Check-sum 法と言う見出しの圧縮法が便利であり、この実験ではそれを用いている。これは一連の文字を適当な長さに区切り、単純に数として加え合せ、それを見出しとする方法である。各単語は単語辞書を引いて検索する。単語は英和辞書のように見出し・品詞・訳語を含んでいなければならない。1つの単語は文中の位置や働きによっていろいろな品詞となり1つとは限らない。また、訳語も多ければ多い程、訳文の質が向上するが、品詞の決定方法と選択方法が現在の機械翻訳の問題点の1つとなっている。見出しの配列もアルファベット順にするか、頻度の高いものを最初にもってくるかによって、単語の検索時間に大きな影響を与える。



図1 機械翻訳の手順

次の構文分析は次のような方法で行われる。まず、あらかじめある言語理論に基づいた文法規則を与えておき、入力文の記号列をしらべて文法規則が適用できるものがあるかどうかを調べる。条件が満足されていると、その規則で与えられる置換、統括などの一連の処理を行ない、以下同様の操作をくり返す。この場合、規則の適用に制限を設け、適用の優先順位を与えたり、その適用を文頭あるいは文末から行なう方法がある。ここでは直接構成要素理論に基づいた隣接結合方式を用いた¹⁾。

次に訳文の生成であるが、構文分析が行われた後には、それに対する訳文は半ばでき上がっている場合が多い。訳文の語順とか、日本語特有の助辞、補助語の挿入なども構文分析に用いる文法規則により与えられているのが普通である。このように訳文が組み立てられていると、それらの情報によって訳語辞書を引き、出力する訳文の形に変換すればよい。

次に訳文の打ち出しでは、組み立てられた訳文をどのような形で出力するかが問題となるが、これは原文の読み込みのときと同様に使用する出力装置によって異なってくる。ここでは結果をローマ字で出力し、読み易くするために分ち書きをさせている。

3. 隣接結合方式

この方式を要約すると、文章中の各単語にそれぞれ単語辞書により品詞記号を与える。品詞記号については前後の脈絡関係を考慮した文法パターンを用意する。この文法パターンはAとBに大別される。Aパターンは隣接する強い結びつきをもった品詞列から構成され、たとえば冠詞+形容詞+名詞、助動詞+動詞、前置詞+名詞など周囲の状況を知ることなく結合できるものがこれに属する。Bパターンは動詞を主とするもので、動詞+名詞、名詞+動詞、関係詞などを含むものである。文章の分析にあたっては、その文章全体にわたって、まずAパターンを適用するが、Aパターンをさらに、3品詞列と2品詞列にわけて、A-3パターン、A-2パターンとよび、

4. 桁移動と PACK, UNPACK 操作

FÖRTRAN によって記号処理を行うことは本質的に無理であるという意見もある²⁾。しかし上述の隣接結合方式による機械翻訳のプログラミングでは、記憶装置の1語の桁移動と、1語のデータを分解したり、また、数語のデータを1語に詰め込む操作を FÖRTRAN のステートメントで表現することができるならば FÖRTRAN による機械翻訳のプログラミングが可能になる。ただし、この部分についてはハードウェアの性質が入ってきて、一般に互換性は保たれない。

図4(a)に示すように、桁移動は1語の中で、ある桁の値を移動させるものである。また、1語のデータを分解して数語のデータとすることを、ここでは UNPACK 操作、逆に数語のデータを1語に詰め込むことを PACK 操作と称し図4(b)に示してある。

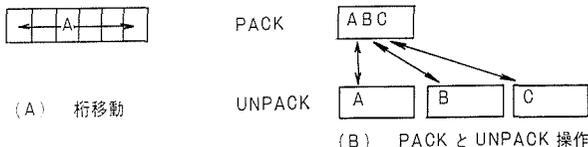


図4 桁移動と PACK, UNPACK 操作

これらの操作は FÖRTRAN のステートメントにはなく、機械語で処理されるのが普通である。そして、これらの操作は膨大なデータを扱う英文和訳のような情報処理の際に、限られた記憶容量を有利に用いる手段としても重要性をもってくる。実際に使用した NEAC2203-G 電子計算機は1語が12桁の10進固定語長方式である。また FORTRAN の A変換で文字を読み込ませると、1文字が2桁の数字で表わされ、左から詰められて、1語に6文字まで格納される。6文字以下の場合には、右方に余白コードを示す99がつけられる。PACK, UNPACK 操作の基礎となる桁移動は、この機種が10進法を採用していることから、(1)式に示すステートメントで行わせることができる。ここで乗算ならば左方へ、除算ならば右方への桁移動が行われる。

$$M \Delta 10^{**} N \dots \dots \dots (1)$$

M: 変数あるいは定数 N: 移動させる桁数
 Δ : 左への移動のとき*, 右への移動のとき/

図5にはMという変数名に入っているAという文字データが右へ10桁移動し、さらに左へ5桁移動させたときの FORTRAN による表現と、そのときのデータの内部表現を示してある。隣接結合方式では、たとえば3品詞列は図6に示す形になる。

| | |
|------------------|--------------|
| M | 139999999999 |
| M/10**10 | 000000000013 |
| (M/10**10)*10**5 | 000001300000 |

図5 桁移動による内部表現

| | | | | | |
|------|------|------|------|------|------|
| J(1) | J(2) | J(3) | J(4) | J(5) | J(6) |
| AA | 101 | BB | 102 | CC | 103 |

図6 品詞記号と訳語番号例

この列の中から AA, BB, CC の3個の品詞記号を1語に詰め込み、それがパターン中にあるかを見なければならぬ。NEAC2203-G で3品詞列を1語に PACK するには、先に述べたこの機種の性質と、整数型演算の特質を考え、(2)式を利用すればよい。同様に BB, CC の2品詞列を作り出す場合には(3)式の表現になる。図6においては L = 6 である。

$$M = (J(L-5)/10^{**}8) * 10^{**}8 + (J(L-3)/10^{**}8) * 10^{**}4 + J(L-1)/10^{**}8 \dots \dots \dots (2)$$

$$N = (J(L-3)/10^{**}8) * 10^{**}8 + (J(L-1)/10^{**}8) * 10^{**}4 + 9999 \dots \dots \dots (3)$$

なお、PACK 操作は訳文の打ち出しの際にも用いている。逆の UNPACK 操作は訳語情報から語順と補助語の情報を分解するときや、原文の読み込みの際などに用いている。図7には K(1) という12桁の数字を2桁ずつ I(1)~I(6)に UNPACK

```

K(1)=123456789012
DO 5 M=1,6
I(M)=k(M)/10**(2*(6-M))
K(M+1)=K(M)-I(M)*10**(6-M)
5 CONTINUE
    
```

図7 UNPACK 操作例

させるプログラムの例を示してある。

5. FÖRTRAN と機械語の比較

前述の様に PACK, UNPACK 操作が FÖRTRAN で可能になったとしても、そのためにプログラムの語数が非常に多くなったり、また演算時間が長くなるようでは、これらの操作を FÖRTRAN で表現しても実際上の意味がなくなる。そこで、PACK, UNPACK 操作の部分を FÖRTRAN で組んだ場合と、それらの部分を機械語でコーディングした場合とのプログラムの語数と演算時間の比較を行ってみた。図8は PACK 操作、図9は UNPACK 操作について測定に用いたプログラムを示してあり、それぞれ (a) は FORTRAN ですべてのプログラムを組んだもの、(b) は PACK, UNPACK 操作の部分に機械語を用いたものである。なお、PACK

操作は3語を1語に詰め込むものであり、UNPACK 操作は1語を6語に分解するのである。図8は適当な22個のデータを与え、最初は20, 21, 22番目のデータを1語にPACKし、次には19, 20, 21番目のデータをPACKするというように、同様の操作を20回くり返すようになっている。(a), (b)とも入力データは同じであり、演算時間の測定の際に、データの読み込み時間が含まれないようにPAUSEステートメントを入れて一旦停止させ、そこから時間を測定するようにしている。(a)の第6行に対応する部分が、(b)では第7行であり、PACK操作を行う機械語のプログラム(MIKUMI)を呼び出して実行するようになっている。図9はプログラム中に与えられている数字12桁のデータを2桁ずつ6個に分解する操作を20回くり返すプログラ

```

1  DIMENSION J(25)
2  READ 10, (J(L),L=1,22)
3  PAUSE
4  L=22
5 40 IF (L-3) 31,30,30
6 30 M=(J(L-2)/10**8)*10**8+(J(L-1)/10**8)*10**4+J(L)/10**8
7  TYPE 20,M
8  L=L-1
9  GO TO 40
10 31 STOP
11 10 FORMAT(22A6)
13  END

```

(A) FORTRAN 使用

```

1  LIBRARY MIKUMI(30)
2  DIMENSION J(25)
3  READ 10, (J(L), L=1, 22)
4  PAUSE
5  L=22
6 40 IF (L-3) 31, 30, 30
7 30 CALL MIKUMI (#J(L), J(L-1), J(L-2), M)
8  TYPE 20,M
9  L=L-1
10 GO TO 40
11 31 STOP
12 10 FORMAT(22A3)
13 20 FORMAT(/A6)
14  END

```

(B) 一部機械語使用

図8 PACK 操作

```

1  DIMENSION I(10), K(10)
2  KAISU=0
3 50 KAISU=KAISU+1
4  IF (KAISU-20) 30, 30, 31
5 30 K(1)=123456789012
6  DO 40 M=1,6
7  I(M)=K(M)/10**(2*(6-M))
8 40 K(M+1)=K(M)-I(M)*10**(2*(6-M))
9  TYPE 20, (I(M), M=1,6)
10 GO TO 50
11 31 STOP
12 20 FORMAT(/6I4)
13  END

```

(A) FORTRAN 使用

```

1  LIBRARY GOJUN3(30)
2  DIMENSION I(30)
3  KAISU=0
4 50 KAISU=KAISU+1
5  IF (KAISU-20) 30, 30, 31
6 30 K1=123456789012
7  CALL GOJUN3(#K1, I(1), I(2), I(3), I(4), I(5), I(6))
8  TYPE 20, (I(M), M=1,6)
9  GO TO 50
10 31 STOP
11 20 FORMAT(/6I4)
12  END

```

(B) 一部機械語使用

図9 UNPACK 操作

ムである。(a)の第6行～第8行では図7と同様の UNPACK 操作が行われ、(b)においてこれに対応する行は第7行であり、UNPACK 操作をする機械語のプログラム (GOJUN3) を呼び出し実行するようになっている。表1にはこれらのプログラムについて、演算時間と語数の測定結果を示してある。測定に用いたどのプログラムにも、結果の打ち出し時間が含まれている。実際の

表1 FÖRTRAN と一部機械語を使用した場合の比較

| | PACK 操 作 | | UNPACK 操 作 | | 実際のプログラム | |
|---------------|----------|----------|------------|-----------|----------|-----------|
| | 語 数 | 演算時間 | 語 数 | 演算時間 | 語 数 | 演算時間 |
| FÖRTRAN 使 用 | 86 | 8.4 sec. | 79 | 17.7 sec. | 880 | 18.5 sec. |
| 一 部 機 械 語 使 用 | 59+15 | 5.6 sec. | 60+22 | 14.1 sec. | 770+64 | 13.7 sec. |
| FÖRTRAN/機 械 語 | 1.16 | 1.50 | 0.96 | 1.26 | 1.06 | 1.35 |

語数欄の『+』の後の数字は用いた機械語の語数である。

プログラムでは、PACK, UNPACK 操作を用いて構文分析を行うものであり、その部分に用いている機械語のプログラムは、PACK, UNPACK 操作に用いたものであるが、さらに2品詞列に対しての処理を行なう同じような機械語のプログラムを使用している。なお、このプログラムの入力データは、パターンが35個であり、平均して13個の品詞記号と訳語番号からなる英文を7つ読み込み、それから作られる2品詞列、3品詞列とパターンのlook-up、さらに補助語の挿入、語順の変更などを行ない、訳語番号の形で出力するものである。他のプログラムと同様、パターンの読み込み時間は含まれていないが、7つの英文の読み込み時間は含まれている。しかし、この時間はほとんど無視できる。すべて FÖRTRAN でプログラムを組んだ場合と一部に機械語を用いた場合との比を見てもわかるように、プログラムの語数はほとんど同じであり、演算時間の比もせいぜい1.5倍程度であるから、実用には影響がないと言える。

6. 翻 訳 実 験

ここで作成した英文和訳のプログラムは ALCÖN II で書かれているが、このプログラムに用いられる部分については標準の FÖRTRAN とほとんど相違はない。プログラムではあらかじめパターン表、単語辞書、訳語辞書を読み込んでおいてから、翻訳されるべき英文の読み込みを行い、STOP ステートメントを用いず、入力される英文がなくなるまで各英文の処理を連続的にくり返すようになっている。DIMENSIÖN ステートメントの寸法は、NEAC2203-G の内部記憶装置にとれる範囲で宣言しており、外部記憶装置は用いていない。プログラム語数は1090語であり、処理の複雑さを考えるとかなり少ない語数であるといえる。

このプログラムの主な特徴をあげると、次の5項目になる。

(1) 扱うデータは人間にとってわかりやすく、かつ、FÖRTRAN での処理を簡単にできるようにしてある。

(2) FÖRTRAN の基本的ステートメントのみを用い、簡単な手直しにより、他機種での使用が可能である。

(3) FÖRTRAN のステートメントで桁移動、PACK 操作、UNPACK 操作をしている。

(4) 2次元の配列を用いたり、あるいは品詞記号と訳語番号を同一の配列で扱ったりして同じような処理をまとめた。

(5) 単語や入力英文の長さ、パターンや辞書の語数が多くなっても、そのまま拡張が可能である。

このプログラムを他機種で使用するには、その機種の1語の構成法を知り、PACK 操作、UNPACK 操作を行なうステートメントを(1)式を用いて書き換えることと、余白を示すコード

(NEAC2203-G では 99) の部分をその機種 of 余白コードにすることだけでよい。あとは、その機種に特有の制限や FÖRTRAN コンパイラの性質に留意すればよい。

7. 翻 訳 実 験 例

実験の規模はあまり大きくなく、パターンは約 120 組、単語辞書は約 160 個の見出しからなり、それに対応する訳語辞書は、約 190 語からなっている。隣接結合方式による構文分析は、パターンだけ与えておけば、入力文の構造に制限をもたないが、実験での入力英文は肯定文と命令文のみとし、疑問文、否定文などのパターンは作成していない。また、肯定文においても、英語の基本的な約 50 文型をえらび、それらの代表的な英文についてパターンを作成し、翻訳させている。

翻訳された文章の一例を入力英文とともに表 2 に示してある。表 2 では、入力英文をさきに示し、その翻訳された結果を次の行に示してある。翻訳文の読み方は、() 内の文字は人間が判断し、読んだり、読まなかったりする。また、「-」の記号は訳語が続いていることを示している。全般的にあまりおかしい訳文にはならなかったが、1つの単語に1つの訳語しか与えなかったために、同じ単語が別な文章ではおかしい訳になっているものもある。また多機能をもつ単語を含んだ英文では、どうしてもすっきりした訳文にならない。たとえば、(5) の「made」は「作る」という訳語だけではなく、また、文章によってはいろいろな働きをする語であるためにどうしても表 2 のような訳文になる。前置詞の「on」という単語も (19) では良い訳であるが、(15) では適当な訳語ではない。この場合、構文上で考えると同じであるためにこのような結果になっている。同じようなことが (14) と (26) の「for」という前置詞に対してもいえる。なお、この実験での翻訳時間は、訳文の打ち出しまで1つの英文につき 10 秒前後である。プログラミングに際しては、各処理ごとにプログラムを作り、それを順次つなぎ合わせて実験をくり返す方法をとったが最も困難を感じた部分は、単語の文字数がさまざまであることであった。したがって、単語の読み込みと訳文の打ち出しの処理に多くの時間がかかっている。また、入力媒体に紙テープを使用しているためデータの修正が面倒であり、パターンの優先順位を決めたり、他のパターンとの関係を考えていたりするには、どうしても試行錯誤的な方法が必要で、そういう面のプログラミングに相当の時間を要している。

8. 結 言

以上述べたごとく、この実験結果から次のことが明らかにされた。

(1) 実験の規模は小さいものであったが、FÖRTRAN による英文和訳が可能であることを確認した。

(2) FÖRTRAN と機械語を比較した場合、プログラムの語数はほとんど変わらず、演算時間の比も実用上はまず影響がないといえる。

(3) 多義語を取り扱う場合には、意味論など、単なる構文分析をこえる他の方法も考慮に入れなければならない。

表 2 翻 訳 例

- (1) BIRDS SING .
TORI GA UTA -(R)U .
- (2) HE IS A BOY .
KARE WA 1 -NO SHOUNEN DE AR -(R)U .
- (3) I LIKE HIM .
WATASI WA KARE O KONOM -(R)U .
- (4) I GAVE HIM A BOOK .
WATASI WA KARE NI 1 -NO HON O ATAE -(I)MASITA .
- (5) I MADE HIM HAPPY .
WATASI WA KARE NO KOUFUKU O TUKUR -(I)MASITA .
- (6) WE CONSIDER IT WRONG TO TELL A LIE .
WAREWARE WA 1 -NO USO O I -(R)U KOTO O WARU -I TO OMO -(R)U .
- (7) HE HAD HIS BOOK STOLEN IN THE TRAIN .
KARE WA SO -NO KISHA NONAKANI KARE NO HON O NUSUM -(R)ARE -(I) MASITA .
- (8) THEY KNOW THAT HE IS HONEST .
KARERA WA KARE WA SHOUJIKI DE AR -(R)U KOTO O SIR -(R)U .
- (9) WE THINK IT DANGEROUS HIS CLIMBING THE MOUNTAIN ALONE .
WAREWARE WA KARE GA HITORIDE SO -NO YAMA O NOBOR -(R)U KOTO KIKEN TO KANGAE -(R)U .
- (10) I SAW HIM LAUGHING .
WATASI WA KARE GA WARU -(R)U NOO MI -(I)MASITA .
- (11) THE ACCIDENT HAS TAUGHT ME THAT RECKLESS PLAN IS DANGEROUS .
SO -NO JIKO WA WATASI NI MUBOU NA KEITAKU WA KIKEN DE AR -(R)U KOTO O OSIE -(I)MASITA .
- (12) I KNOW HOW TO DO IT .
WATASI WA IKANI SORE O S(U) -(R)U KA O SIR -(R)U .
- (13) THEY TOLD THE GIRL WHEN TO START .
KARERA WA SO -NO SHOUJO NI ITU DEKAKE -(R)U KA O I -(I)MASITA .
- (14) SHE MADE A NEW DRESS FOR HERSELF .
KANOJO WA 1 -NO ATARASI -I YOUFUKU O KANOJOJISIN NOTAMENI TUKU R -(I)MASITA .
- (15) HE SPENT MUCH MONEY ON BOOKS .
KARE WA OOKU NO KANE O HON NOUENI TUKA -(I)MASITA .
- (16) ASK HIM HIS NAME .
KARE NI KARE -NO NAMAE O TAZUNE -(R)U BESI .
- (17) ENCOURAGING A LAZY PERSON IS PROFITLESS .
1 -NO TAIMAN NA NINGEN O HAGEMAS(U) -(R)UKOTO WA MUDA DE AR -(R)U .
- (18) TO SEE IS TO BELIEVE .
MI -(R)UKOTO WA SINJI -(R)UKOTO DE AR -(R)U .
- (19) THERE IS A BOOK ON THE DESK .
SO -NO TUKUE NOUENI 1 -NO HON GA AR -(R)U .
- (20) YOU ARE YOUNGER THAN I .
ANATA WA WATASI YORIMO WAKAI .
- (21) WHAT HE SAID IS TRUE .
KARE GA I -(I)MASITA KOTO WA SINJITU DEAR -(R)U .
- (22) THIS IS THE HOUSE THAT HE BUILT .
KORE WA KARE GA TATE -(I)MASITA SO -NO IE DE AR -(R)U .
- (23) THE BOY WAS BITTEN BY A DOG .
SO -NO SHOUNEN WA 1 -NO INU NIYOTTE KAM -(R)ARE -(I)MASITA .
- (24) I HAVE KNOWN HIM FROM HIS CHILDHOOD .
WATASI WA KARE -NO YOUNENJIDAI KARA KARE O SIR -(I)MASITA .
- (25) I HAD ALREADY FINISHED MY WORK WHEN HE CAME .
KARE GA K(U) -(I)MASITA TOKI WATASI WA WATASI -NO SIGOTO O SU DENI OE -(I)MASITA .
- (26) IT IS DIFFICULT FOR HIM TO READ THE BOOK .
SO -NO HON O YOM -(R)U KOTO WA KARE NOTAMENI KONNAN DE AR -(R)U .

参 考 文 献

- 1) A. D. Booth: Machine Translation, North-Holland Publishing Co. (1967)
- 2) 中村忠之: 言語の機械翻訳, 北大工学部電子工学科卒業論文 (1968)
- 3) 坂井利之: 翻訳するコンピューター, 講談社 (1969)
- 4) 坂井, 長尾: 機械による英文和訳, 電気通信学会誌 Vol. 49, No. 2 (1966)
- 5) 坂井, 杉田, 渡辺: 電子計算機による和文英訳, 情報処理 Vol. 10, No. 6 (1969)
- 6) 西村恕彦: なぜ FÖRTRAN を止めたか, 数理科学 Vol. 6, No. 12 (1968)
- 7) 中村, 枅内, 仲丸: FÖRTRAN による機械翻訳プログラムについて, 電気四学会北海道支部連合大会講演論文集 (1969)
- 8) C. C. Fires: The Structure of English, Longmans, Green and Co Ltd. London (1965)
- 9) P. Roberts: Patterns of English, Harcourt Brace & World, Inc. (1956)
- 10) 坂井, 長尾: 最近の言語処理研究について, 情報処理 Vol. 10, No. 1 (1969)
- 11) W. L. Price: Computer translation-is it worthwhile, Electronics & Power, Vol. 13 September (1967)
- 12) 西村恕彦: 機械翻訳のための英和文法の研究, 電気試験所研究報告第 696 号 (1969)